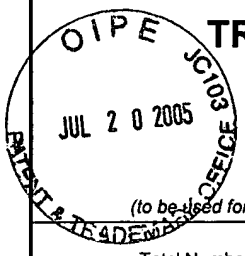
PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.



TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission: 13

Application Number	09/934,738
Filing Date	August 22, 2001
First Named Inventor	Molnar, Ingo
Art Unit	2135
Examiner Name	John F. Gianola
Attorney Docket Number	019322-000340

ENCLOSURES (check all that apply)

- ☐ Fee Transmittal Form
  - ☐ Fee Attached
- ☐ Amendment / Reply
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Documents
- ☐ Response to Missing Parts/ Incomplete Application
  - ☐ Response to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) _____
  - ☐ Landscape Table on CD
- ☐ After Allowance Communication to Group
- ☒ Appeal Communication to Board of Appeals and Interferences
- ☐ Appeal Communication to Group *(Appeal Notice, Brief, Reply Brief)*
- ☐ Proprietary Information
- ☐ Status Letter
- ☐ Other Enclosure(s) *(please identify below):*
  Reply Postcard
  Check in the amount of $500

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	MOORE & VAN ALLEN		
Signature			
Printed Name	STEVEN B. PHILLIPS		
Date	July 18, 2005	Reg. No.	37,911

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below.

Signature	Patricia L. Pontello		
Typed or printed name	Patricia L. Pontello	Date	July 18, 2005

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Molnar	
Serial No.: 09/934,738	Confirmation # 9016
Filing Date: August 22, 2001	
Docket No. 019322-000340	
Examiner: Gianola, John F.	
Group Art Unit: 2145	
Title: Embedded Protocol Objects	

Commissioner for Patents
Washington, DC 20231
BOX AF

APPEAL BRIEF

Real Party in Interest

Red Hat, Inc. is the real party in interest.

Related Appeals and Interferences

There are no other appeals or interferences, known to the Appellant, or Appellant's legal representatives, which will directly affect or be directly affected by or have a bearing on the Board's decision in this pending appeal.

Status of Claims

Claims 1-14 are pending. The May 20, 2005 final rejection of all pending claims (claims 1-14) is being appealed herein.

Status of Amendments

There were no amendments filed after the final office action of May 20, 2005. Applicants chose to proceed directly with this appeal. All previous amendments have been entered.

Summary of Claimed Subject Matter

The claimed invention improves the performance of data-serving applications by providing a mechanism for dynamic and static protocol objects to be mixed together at a server and included in a dynamic reply to a communication request made by a client application. In effect, static protocol objects can be embedded in dynamic replies, along with dynamic protocol objects. If a reply, such as a web page, requires content, only a small portion of which is dynamic, the reply can be created and sent using mostly cached, static protocol objects, resulting in the reply being sent more quickly and efficiently than is possible with prior-art server systems.

The invention allows a communication server to respond to an application protocol request received from a client application by creating at least one dynamic protocol object to form at least a portion of the reply. The reply is designed to be able to include embedded, static protocol objects. The dynamic protocol object or objects are sent to the client application. Static protocol objects are then retrieved and sent to the client application to complete the reply to the application protocol request at the client application. The static protocol object or objects are effectively embedded in the dynamic reply.

Claims 1, 5, 9 and 11 are the independent claims. Claim 9, one of the independent claims, is the only claim with means-plus-function elements. For purposes of complying with 37 C.F.R. § 41.37(c), support for the recitations in these claims is discussed below.

Claim 1

Claim 1 recites a method of responding to a client application. The elements of claim include a first element of "receiving from the client application an application protocol request corresponding to a response that can be displayed as a combination of a dynamic protocol object and a static protocol object." Support for this element occurs in Figure 3, block 301 and in the specification at paragraph [0027].

The second element of the claim is "creating at the server the dynamic protocol object." Support for this element occurs in Figure 3, block 303 and in paragraph [0028] of the specification.

The third element of the claim is "sending the dynamic protocol object to the client application." This element is supported by in Figure 3, block 304 and paragraph [0028] of the specification.

The fourth element of the claim is "retrieving the static protocol object from a cache disposed in an operating system kernel." This element is supported by Figure 3, block 305; Figure 4, blocks 405, 406 and 407; and paragraphs [0028], [0032] and [0034] of the specification.

The fifth element of the claim is "sending the static protocol object to the client application." This element is supported by Figure 3, step 306 and by paragraph [0028] of the specification.

Claim 5

Claim 5 recites a computer program on a medium with computer program code embodied therein. Computer program products are discussed in paragraph [0041] of the specification and shown in Figure 6. The elements of claim 5 are commensurate with those of claim 1 and support for the rest of the recitation in each element can be found in the same places as discussed above with regard to claim 1.

Claim 9

Claim 9 recites an apparatus for responding to a client application. The first element of Claim 9 is a cache disposed in an operating system. Support for this element occurs in Figure 1, block 105 and in paragraph [0022] of the specification. The remainder of the elements of claim 9 are recited in means-plus-function language pursuant to 35 U.S.C. § 112 ¶ 6. In each case, the means for performing the function is the combination of a computer system like that shown in Figure 6 and discussed in paragraphs [0020] and [0025] of the specification and a computer program product as discussed in paragraph [0019] of the specification. The function performed on each element of claim 9 corresponds to one of the elements of claim 1 and support for each function can be similarly found.

Claim 11

Claim 11 recites an instruction execution system operable as a communication protocol server. The instruction execution system is supported by paragraph [0041]

and Figure 6. The claim elements of claim 11 correspond to those of Figure 1 and support can be found in the same places.

Grounds of Rejection to be Reviewed on Appeal

1. Claims 1, 3, 5, 7, 9 and 11-14 were rejected for purportedly lacking novelty under 35 U.S.C. § 102(b) in view of Iyengar et al., "An Analysis of Web Server Performance."

2. Claims 2, 4, 6, 8, 10 were rejected as purportedly obvious under 35 U.S.C. § 103(a) in view of Iyengar et al. in combination with Pai et al. "IO-Lite: A unified I/O Buffering and Caching System."

Arguments

Claims 1, 3, 5, 7, 9 and 11-14 are novel under 35 U.S.C. § 102(b).

The examiner has rejected claims 1, 3, 5, 7, 9, and 11-14 under 35 U.S.C. § 102(b) in view of an article by Iyengar et al. entitled "An Analysis of Web Server Performance." In order for a claim to be anticipated, the cited reference must teach every element of the claim, either expressly or inherently. MPEP 2131. The Examiner has cited page 1944, column 2, lines 6-19 of the Iyengar article as disclosing "web servers transmitting pages that contain both static and dynamic content." But, the examiner confuses requests for pages, files and objects. Iyengar discloses a mechanism by which (1) an application requests a dynamic page, (2) the server responds with a dynamically generated protocol object which refers to (but does not contain) a "static file" and (3) the application sends a separate request for the static file. This is demonstrated on p. 1944 of Iyengar, lines 17-19: "Sites which generate close to

100% of all Web pages dynamically often receive well over 50% of all requests for static files (usually images) included within the dynamic pages." Clearly, the requests for static content are generated separately. Iyengar's reference to static files "included" in dynamic pages recognizes that when *displayed*, the page may have both static and dynamic content, not that any individual response from the web server includes both the static and dynamic content.

Under Iyengar, an application may create a single page by sending multiple requests, each for a static or a dynamic element of the page's content. Iyenger thus does not disclose an application sending a single request and the server responding with both static and dynamic content. Applicant's independent claims 1, 5, 9 and 11 recite "a response that can be displayed as a combination of a dynamic protocol object and a static protocol object." This recitation is incorporated in Applicant's dependent claims 3, 7 and 12-14 through their dependency from amended claims 1, 5, 9, 11. Applicant submits that all of his present claims are patentably distinguishable from Iyengar for at least this reason. Applicant therefore re-asserts claims 1, 3, 5, 7, 9 and 11-14 as novel and hereby appeals the rejection of these claims, both collectively and individually.

In addition to the above, Applicant notes that with respect with at least claim 9, the Examiner has cited page 1946, column 1, lines 8-12 of Iyengar as disclosing "a cache for frequently accessed dynamic pages." The cache as disclosed in claimed Applicant's application is of protocol objects, not complete web pages.

Applicant's claims are further distinguishable from Iyengar, as Iyengar discusses caching *dynamic* pages to avoid having to run a separate program to recreate them,

whereas Applicant's claims recite caching *static* protocol objects to avoid, among other things, the cost of context switching. Furthermore, all of Applicant's claims recite that the cache is disposed within an operating system kernel. Iyenger is silent on this feature. The Examiner has cited to Pai et al. as teaching, in conjuction with Iyenger, using a kernel cache as part of a web server. (See Pai et al.: page 47, lines 35-36; page 48, lines 11, 31-35; and page 49, lines 3-10 and 27-28.) However, Pai et al. teaches caching the raw file data, not the protocol object(s) generated from that file – the entire thrust of the article is keeping file data in the kernel until it is transmitted.

Applicant asserts an additional reason that commensurate claims 13 and 14 are novel. Claims 13 and 14 recite that the cache is a protocol object cache. The cache disclosed in Iyengar and Pai et al. are not protocol object caches but caches of either web pages (in Iyengar) or raw files (in Pai et al.). Protocol objects differ significantly from both web pages and raw files at least in that protocol objects embody information required by the underlying protocol in addition to the raw data being transmitted. As a result, this information does not need to be re-created for each transmission, as it would need to be by Iyengar or Pai et al. Thus claims 13 and 14 are novel for at least this additional reason.

Claims 2, 4, 6, 8 and 10 are non-obvious under 35 U.S.C. § 103(a).

The Examiner has rejected claims 2, 4, 6, 8 and 10 under 35 U.S.C. § 103(a) as being obvious in view of Iyengar in combination with an article by Pai et al., namely "IO-Lite: A Unified I/O Buffering and Caching System." In order to establish the obviousness of a claim, one of the showings the Examiner must make is that all the

claim limitations are taught or suggested by the prior art. MPEP 2143.03. The Examiner relies on Iyengar as previously discussed to disclose the concept of mixing static and dynamic content. Thus, the arguments recited above are also applicable to the rejection under section 103, namely, that Iyengar does not disclose forming a complete response to a request out of *both* static and dynamic objects as claimed. There is no teaching related to these recitations in Pai either; therefore the rejection under section 103 cannot stand. Further, Pai only discloses caching of file data, not of protocol objects. Under Pai, the protocol object will need to be re-created for each request for the same static file. Thus, Applicant maintains that claims 2, 4, 6, 8 and 10 cannot be obvious in view of Iyengar in combination with Pai et al. Applicant therefore re-asserts claims 2, 4, 6, 8 and 10 as non-obvious and hereby appeals the rejection of these claims, both collectively and individually.

Claims 2, 6 and 10 are not obvious for the additional reason that they recite the cache as being a protocol object cache. Neither Iyengar nor Pai et al. disclose such a cache. Iyengar's cache is of dynamic web pages and Pai et al.'s is of raw files.

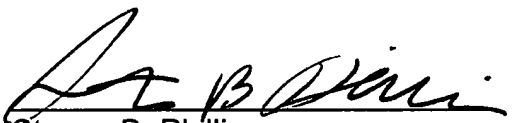
Conclusion

For the reasons state above, the Applicant respectfully submits that all of the rejections standing in this application are improper. With regard to claims 1, 3, 5, 7, 9 and 11-14, the Examiner has failed to establish a *prima facie* lack of novelty under 35 U.S.C. § 102(b). With regard to claims 2, 4, 6, 8 and 10, the Examiner has failed to establish a *prima facie* case of obviousness under 35 U.S.C. § 103. Therefore,

Applicant respectfully submits that claims 1-14 are in condition for allowance.

Accordingly, the reversal of the rejections of these claims is respectfully requested.

Respectfully submitted,

Date: 18 July 2005 By: 

Steven B. Phillips
Registration No. 37,911,
Attorney for Applicants

Moore & Van Allen PLLC
P.O. Box 13706
Research Triangle Park, NC 27709
Telephone: (919) 286-8000
Facsimile: (919) 286-8199

Appendix

The following is a clean copy of the claims involved in this appeal.

1. In a communication server, a method of responding to a client application, the method comprising the steps of:

receiving from the client application an application protocol request corresponding to a response that can be displayed as a combination of a dynamic protocol object and a static protocol object;

creating at the server the dynamic protocol object;

sending the dynamic protocol object to the client application;

retrieving the static protocol object from a cache disposed in an operating system kernel; and

sending the static protocol object to the client application.

2. The method of claim 1 wherein the cache disposed within the operating system kernel is a protocol object cache.

3. The method of claim 1 wherein the application protocol request and the reply are formatted according to a hypertext transmission protocol (HTTP).

4. The method of claim 2 wherein the application protocol request and the reply are formatted according to a hypertext transmission protocol (HTTP).

5. A computer program product comprising a medium having computer program code embodied therein, the computer program code for enabling a server to respond to a client application, the computer program code comprising:

instructions for receiving from the client application an application protocol request corresponding to a response that can be displayed as a combination of a dynamic protocol object and a static protocol object;

instructions for creating at the server the dynamic protocol object;

instructions for sending the dynamic protocol object to the client application;

instructions for retrieving the static protocol object from a cache disposed in an operating system kernel; and

instructions for sending the static protocol object to the client application.

6. The computer program product of claim 5 wherein the cache disposed within the operating system kernel can be a protocol object cache.

7. The computer program product of claim 5 operable to format the application protocol request and the reply according to a hypertext transmission protocol (HTTP).

8. The computer program product of claim 6 operable to format the application protocol request and the reply according to a hypertext transmission protocol (HTTP).

9. Apparatus for responding to a client application, the apparatus comprising:

a cache disposed in an operating system kernel;

means for receiving from the client application an application protocol request corresponding to a response that can be displayed as a combination of a dynamic protocol object and a static protocol object;

means for creating at a server the dynamic protocol object;

means for sending the dynamic protocol object to the client application;

means for retrieving a static protocol object from the cache through an operable connection to the cache; and

means for sending the static protocol object to the client application.

10. The apparatus of claim 9 wherein the cache can be a protocol object cache.

11. An instruction execution system operable as a communication protocol server, operable to respond to a client application by performing the steps of:

receiving from the client application an application protocol request corresponding to a response that can be displayed as a combination of a dynamic protocol object and a static protocol object;

creating at the server the dynamic protocol object;

sending the dynamic protocol object to the client application;

retrieving the static protocol object from a cache disposed in an operating system kernel; and

sending the static protocol object to the client application.

12. The instruction execution system of claim 11 further operable as a hypertext transmission protocol (HTTP) server.

13. The instruction execution system of claim 11 wherein the cache can be a protocol object cache.

14. The instruction execution system of claim 12 wherein the cache can be a protocol object cache.